Indonesian Journal of Computing, Engineering, and Design

Journal homepage: http://ojs.sampoernauniversity.ac.id/index.php/IJOCED

Emotion Recognition in Javanese Music: A Comparative Study of Classifier Models with a Human-Annotated Dataset

Moh Erwin Septianto[1], Ariana Tulus Purnomo[1], Ding-Bing Lin[2], Chang-Soo Kim[3]*

[1]Computer Science Study Program, Faculty of Engineering and Technology, Sampoerna University, Jakarta Selatan 12780, Indonesia.

[2]Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan.

[3]Department of Information Systems, Pukyong National University, Busan 608737, Republic of Korea.

*Correspondence e-mail: ariana.purnomo@sampoernauniversity.ac.id

ABSTRACT

With advancements in machine learning and the increasing availability of music datasets, Music Emotion Recognition (MER) has gained significant attention. However, research focusing on Indonesian traditional music, particularly Javanese music, remains limited. Understanding emotions in Javanese music is crucial for preserving cultural heritage and enabling emotion-aware applications tailored to Indonesian traditional music. This study investigates the effectiveness of three well-established machine learning models, 1D Convolutional Neural Networks (1D-CNNs), support Vector Machines (SVMs), and XGBoost, in classifying emotions in Javanese music using a manually annotated dataset. The dataset consists of 100 Javanese songs from various genres, including Dangdut, Koplo, and Campur Sari, annotated based on the Thayer emotion model. The models' performance was assessed using different data split ratios, with accuracy rates exceeding 70%. Among the tested classifiers, SVM exhibited the highest and most stable accuracy.

ARTICLE INFO

Article History:
Received 07 Aug 2025
Revised 20 Aug 2025
Accepted 08 Oct 2025
Available online 21 Oct 2025

Keywords:
Classification,
Javanese music,
Machine Learning,
MFCC feature extraction,
Music emotion recognition.

1. INTRODUCTION

Music plays a vital role in conveying emotions and evoking psychological responses (Sloboda & Juslin, 2001). The field of Music Emotion Recognition (MER) has emerged as a significant area within Music Information Retrieval (MIR) that aims to classify and analyze the emotional content of music using computational techniques (Yang & Chen, 2011). While MER has been extensively studied in Western and contemporary music genres, research focusing on Indonesian traditional music, particularly Javanese music, remains sparse. Given Javanese music's unique melodic structures, rhythmic intricacies, and cultural significance, understanding its emotional characteristics presents both opportunities and challenges

Early research primarily relied on handcrafted features extracted from audio signals, such as Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, and tempo (Naveenkumar & Kaliappan, 2019). Traditional machine learning models such as Support Vector Machines (SVM) and Random Forests have been widely used for classification tasks that demonstrate moderate success. More recently, deep learning approaches, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have gained prominence due to their ability to learn hierarchical representations of audio features that can improve classification accuracy (Ansari et al., 2021), (Kaur & Kumar, 2021), (Rezaul et al., 2024).

One major MER advancement is the introduction of end-to-end deep learning models that can directly process raw audio signals without requiring extensive feature engineering. Studies have shown that 1D-CNN architectures effectively capture temporal patterns in music that make them suitable for emotion recognition tasks (Allamy & Koerich, 2021). Ensemble learning techniques, such as XGBoost and hybrid models combining CNNs with Long Short-Term Memory (LSTM) networks, have been explored to enhance prediction performance (Passricha & Aggarwal, 2019), (Nissar et al., 2019). These advancements have significantly improved MER's robustness and accuracy across different musical genres in Western music, where annotated datasets are more readily available.

Javanese music encompasses various genres, from traditional gamelan compositions to modern adaptations such as Dangdut, Koplo, and Campur Sari. These genres blend Indigenous musical elements with contemporary influences to create a rich auditory experience that reflects Indonesia's diverse cultural landscape. However, the lack of publicly available annotated datasets for Javanese music poses a significant hurdle in developing emotion recognition models tailored to this genre.

Despite the advancements of MER, the MER research on non-Western music, including Javanese music, remains underdeveloped. The unique timbral and rhythmic structures of Javanese music pose challenges for existing models primarily trained on Western datasets. Additionally, the subjectivity of emotion perception across different cultural backgrounds makes it difficult to establish universal annotation standards (Scherer, Clark-Polner, & Mortillaro, 2011). Addressing these challenges requires the development of culturally informed datasets and adapting machine learning models to capture the nuances of Javanese music better.

This study aims to bridge this research gap by creating a manually annotated dataset to evaluate the performance of three widely used machine learning classifiers, 1D-CNNs, SVMs, and XGBoost, in classifying emotions in Javanese music. Using the Thayer model (Thayer, 1990), (Russell,

1980), music experts annotated 100 Javanese songs into four basic emotional categories: happy, sad, angry, and relaxed. Although Thayer (1990) and Russell (1980) are classical models, they remain the cornerstone of emotion representation in Music Emotion Recognition research due to their simplicity, interpretability, and enduring relevance in both traditional and contemporary studies. The classification models were trained and tested across different data split ratios to assess their accuracy and effectiveness. The results indicate that all three classifiers achieved accuracy rates above 70%, with SVM demonstrating the most consistent performance across multiple configurations. The findings of this study highlight the importance of annotated Javanese music datasets and appropriate model selection in MER.

2. METHODOLOGY

In this section, we will discuss the flow of this research and the methods used for conducting this experiment in machine learning, including the explanation of data collection, feature extraction, and classification. This research used an Intel i7-12700 with 32GB of RAM and an Nvidia T400 with 4GB of RAM, facilitated by Sampoerna University.

The research was conducted by collecting music songs on YouTube that will be used for annotation, with the restriction that only Javanese music will be used. For more precise boundaries, Javanese music is defined as a song that is sung in Javanese. So, even though the song used a Javanese musical instrument, if the song was not in Javanese, then the song will not be considered a Javanese song. The song was then annotated using Self-Assessment Manikin (SAM), which uses only four emotions (angry, relaxed, happy, and sad) based on the Thayer model (Thayer, 1990), (Russell, 1980). The annotator was chosen based on some criteria. After annotation, the music goes through three stages: preprocessing, augmentation, and feature extraction. Machine learning models are then tested to identify emotions in Javanese music using an audio-based approach.

The overall methodological framework of this study is summarized in **Figure 1**, which consists of three main stages: Emotion Annotation Process, Music Preprocessing, and Classification. In the Emotion Annotation Process, Javanese music data were first collected and organized, after which an appropriate annotation framework (Thayer's two-dimensional arousal–valence model) was selected. A group of qualified annotators was recruited to manually rate each song on arousal and valence dimensions. The raw ratings were then normalized to a [0, 1] scale to ensure comparability across annotators. The normalized values were plotted to visualize the emotional distribution of the dataset, forming the basis of the final emotion labels.

The Music Preprocessing stage involved several signal processing steps to prepare the audio for feature extraction. Each audio file was first converted to mono, then equalized to balance amplitude levels across samples. Subsequently, the songs were segmented into fixed-length clips to capture consistent emotion-related segments. From these segments, one-dimensional acoustic features were extracted, including MFCCs, Chroma features, and spectral descriptors. The resulting features were then encoded and scaled to standardize their numerical range before input into the classification models.

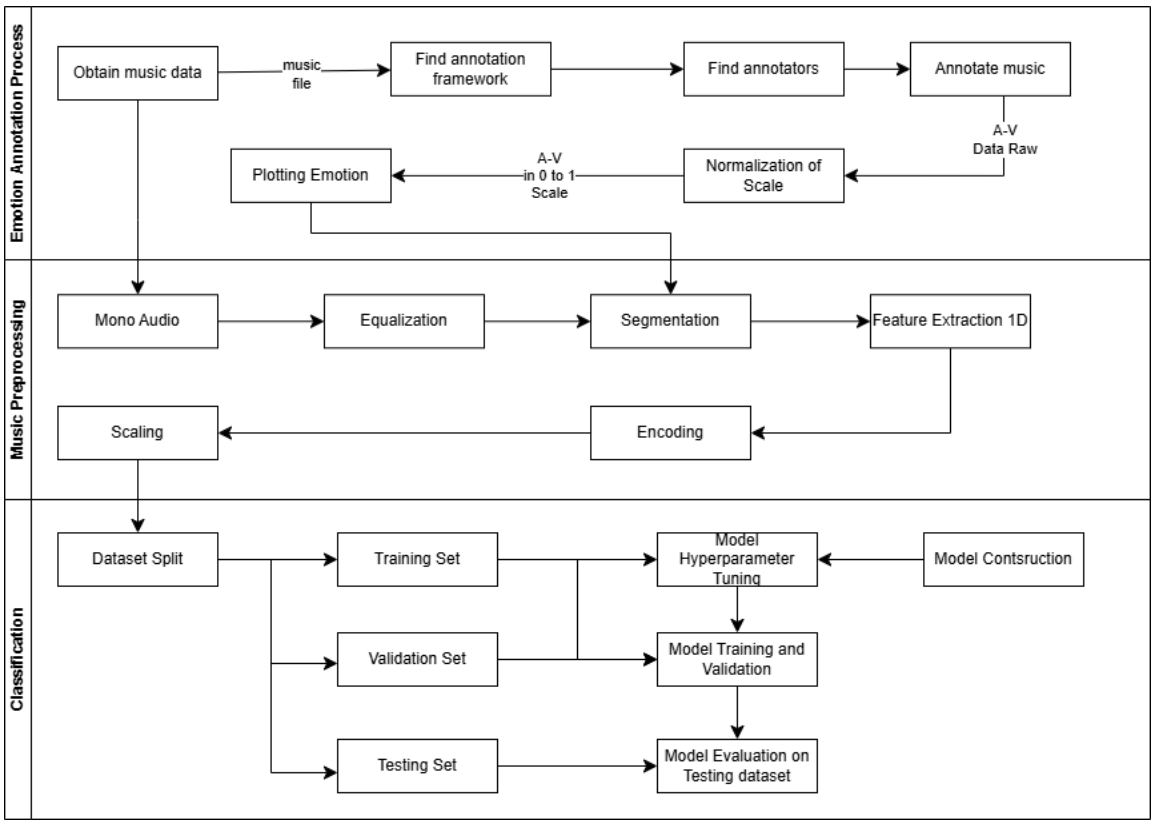

Figure 1. Research Framework

In the Classification stage, the pre-processed and annotated dataset was split into training, validation, and testing subsets. The training set was used to train the models, the validation set for tuning hyperparameters, and the testing set for final performance evaluation. Model construction and evaluation were conducted using three classifiers: Support Vector Machine (SVM), XGBoost, and 1D Convolutional Neural Network (1D-CNN) to compare performance in recognizing emotions from Javanese music.

2.1. Music Preprocessing

All audio files were collected in .wav format, resampled to 44.1 kHz, and converted to mono for consistency. Each song was segmented into 30-second clips using Librosa's built-in utilities. Subsequently, the Short-Time Fourier Transform (STFT) was applied to extract frequency-domain representations, forming the basis for feature extraction. The resulting segments were normalized and labelled according to emotion annotations before being used as input to the machine learning models. The algorithm for converting an audio file to mono is described in **Algorithm 1**.

Algorithm 1: Convert to Mono

Input: audio

Output: mono audio file

Pseudocode

```
1: Initialize `mono_audio` ← None
2: Load audio: `audio` ←
   librosa.load(file)
3: Check number of channels:
      if `audio.channels` > 1 then
            `mono_audio` ← au-
            dio.set_channels(1)
      else
            `mono_audio` ← audio
4: Save `mono_audio` as output file
```

Preprocessing in **Algorithm 1** involved loading each .wav file, converting it to mono, segmenting it, and applying the Short-Time Fourier Transform (STFT) to extract time-frequency domain features. Next, this study divided the music audio into numerous segment files at 30-second

intervals. Any segment fewer than thirty will be left off from the dataset. There will be 984 files altogether across the section. Every segment file expresses the same feeling as music used in past years. Segmentation steps are described in **Algorithm 2.**

Algorithm 2: Audio Segmentation

Input: mono audio

Output: segmented audio file

Pseudocode

```
Initialize `segments` ← empty list
Define `segment_length` ← segment_duration × 1000   # convert to milliseconds
For `i` from 0 to len(mono_audio) step `segment_length` do
    Extract segment: `segment ← mono_audio[i : i + segment_length]
    If len(segment) == segment_length then
        append `segment` to `segments`
Save all `segments` as individual files
```

To validate the integrity and tonal characteristics of each audio file, waveform and Mel-spectrogram visualizations were generated using the Librosa library. These visualizations helped confirm the presence of distinct harmonic and rhythmic patterns typical of Javanese music. **Figure 2** and **3** shows an example of a waveform and its corresponding Mel-spectrogram for a Javanese song.

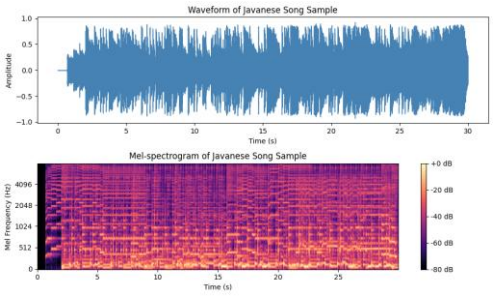

Figure 2. Waveform and Mel-spectrogram of a 30-second segment from the Javanese song.

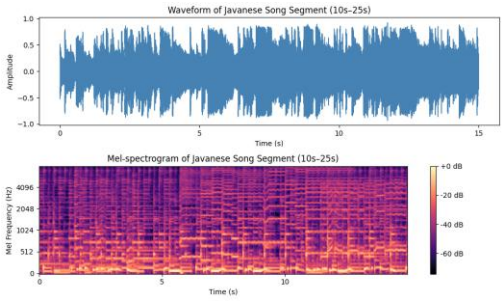

Figure 3. Waveform and Mel-spectrogram of a 15-second segment from the Javanese song.

To provide visual evidence of acoustic differences across emotional categories, **Figure 4** presents waveform and Mel-spectrogram representations for four representative Javanese songs labelled as Happy, Sad, Angry, and Relaxed. Clear spectral and energy distribution differences can be observed across categories. **Figure 4.** Waveform (top) and Mel-spectrogram (bottom) visualizations for four representative Javanese songs labelled as Happy, Sad, Angry, and Relaxed. Each subplot displays the middle 30-second segment of the song. Distinct amplitude and spectral patterns can be observed across emotions. Each emotional category displays distinct time and frequency characteristics. Happy and Angry songs exhibit higher energy and broader spectral content, while Sad and Relaxed songs demonstrate smoother waveforms, lower intensity, and narrower spectral bandwidths. These variations confirm that the extracted features (e.g., MFCCs, Chroma, Spectral Centroid) capture relevant emotional cues consistent with the Thayer model dimensions of arousal and valence.

The overall preprocessing workflow is illustrated in **Figure 5,** which outlines the sequential steps applied to each audio file before model training. First, all songs were collected in raw .wav format and standardized to a 44.1 kHz sampling rate. Each track was then converted to mono to ensure consistent channel representation across the dataset. Next, the audio was segmented into uniform time intervals (0–30

seconds) to capture manageable, emotion-consistent excerpts.

Following segmentation, the Short-Time Fourier Transform (STFT) was applied to convert the signal from the time domain to the time-frequency domain, preserving both spectral and temporal information essential for emotion analysis. From these STFT representations, a set of acoustic features, including MFCCs, Chroma features, Spectral Centroid, Spectral Rolloff, RMS Energy, and Zero-Crossing Rate, was extracted to characterize timbral, harmonic, and rhythmic properties.

Each segment was then associated with its corresponding emotion label derived from the manual annotation process based on the Thayer model. Finally, these labelled feature vectors served as the input to the machine learning classifiers (SVM, XGBoost, and 1D-CNN) for emotion prediction.

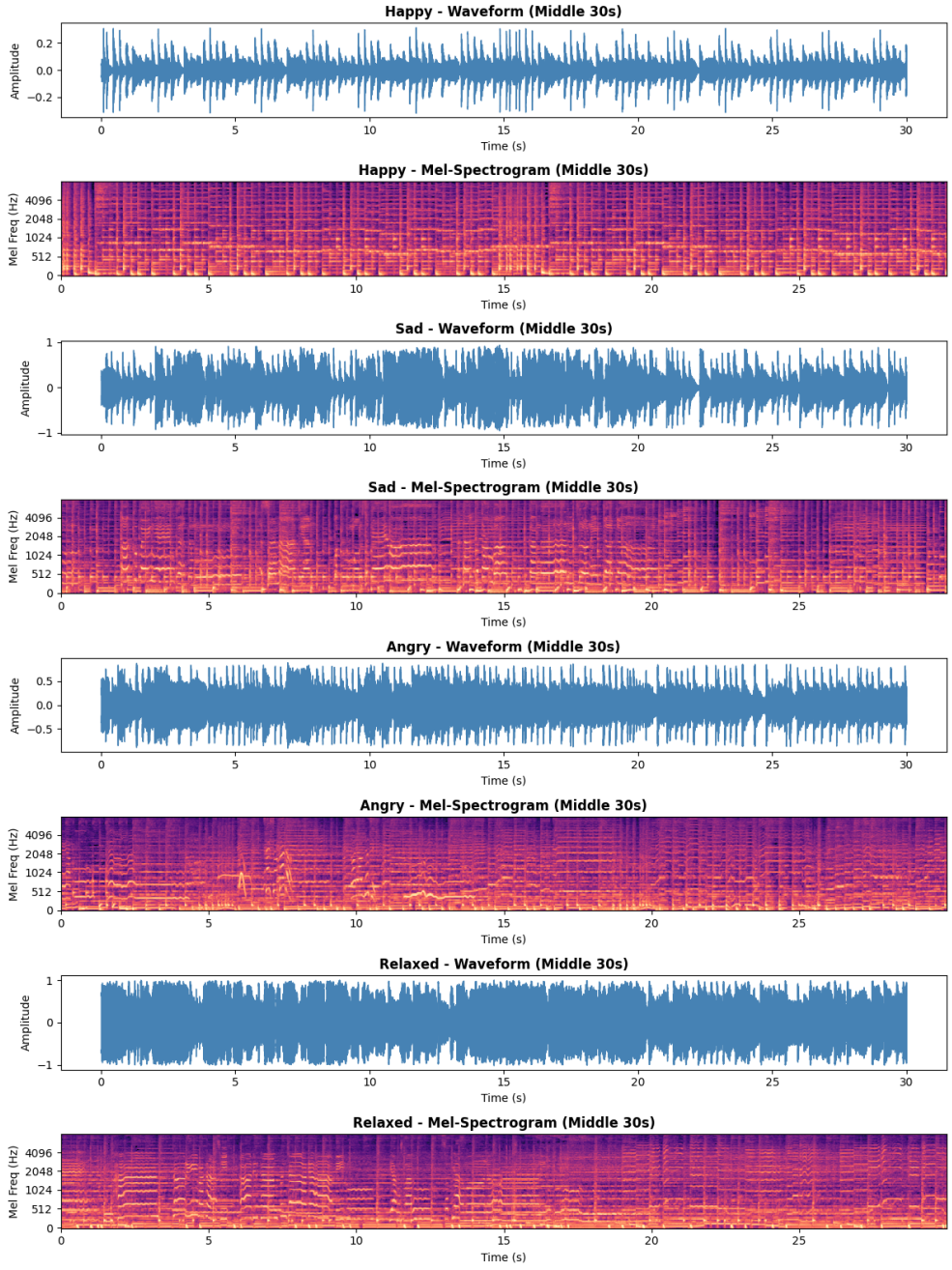

Figure 4. Waveform and Mel-Spectrogram Representations of Javanese Songs Across Four Emotions.

Figure 5. Audio preprocessing illustrating sequential steps from raw audio input to model-ready data.

2.2. Feature Extraction

Converting audio information into relevant representations for emotion identification depends critically on feature extraction. This study captures Javanese music's emotional content through low-level and high-level auditory qualities. Every audio file has features extracted using the Python Librosa package. The key_scale_finder function determines each song's key and scale (major or minor), which will provide valuable information about the song's tonality. Feature extraction was based on STFT-derived representations, including MFCCs, Chroma features, and Spectral Descriptors (Centroid, Rolloff, and Flux), computed using Librosa. The steps are represented in **Algorithm 3**.

Algorithm 3: Key and Scale Finder

Input: 'file'
Output: 'scale', 'key'
Pseudocode
1: Initialize `pitches` ← ['C','C#','D','D#','E','F','F#','G','G#','A','A#','B']
2: Load audio file: `y, sr` ← librosa.load(file)
3: Compute chroma STFT of `y`
4: Find key: `key` ← index of maximum value in chroma STFT
5: Determine scale:
    if `pitches[key]` ends with '#' then
        `scale` ← "minor"
    else
        `scale` ← "major"
6: Return `scale`, `pitches[key]`

The feature_1d function extracts various features, including tempo, RMS energy, chroma features, spectral centroid, spectral roll off, zero crossing rate, tonnetz, Mel-spectrogram, and MFCCs. These steps are described in **Algorithm 4.**

Algorithm 4: Feature Extraction from Audio

Input: 'file'
Output: 'features', '(array)'
Pseudocode
1: Initialize `f` ← empty list
2: Load audio file: `y, sr` ← librosa.load(file)
3: Extract tempo
    3.1: `tempo` ← first value from librosa.beat.tempo(y, sr)
    3.2: append `tempo` to `f`
4: Extract RMS
    4.1: `rms` ← RMS(y)
    4.2: `rms_mean` ← mean(rms)
    4.3: `rms_var` ← variance(rms)
    4.4: append `rms_mean`, `rms_var` to `f`
5: Extract chroma features
    5.1: `chroma` ←chroma_STFT(y)
    5.2: `chroma_mean` ←mean(chroma)
    5.3: `chroma_var` ← variance(chroma)
    5.4: append `chroma_mean`, `chroma_var` to `f`
6: Extract spectral centroid
    6.1: `centroid` ← spectral_centroid(y)
    6.2: `centroid_mean` ← mean(centroid)
    6.3: `centroid_var` ← variance(centroid)
    6.4: append `centroid_mean`, `centroid_var` to `f`
7: Extract spectral rolloff
    7.1: `rolloff` ← spectral_rolloff(y)
    7.2: `rolloff_mean` ← mean(rolloff)
    7.3: `rolloff_var` ← variance(rolloff)
    7.4: append `rolloff_mean`, `rolloff_var` to `f`
8: Extract zero crossing rate
    8.1: `zcr` ← zero_crossing_rate(y)
    8.2: `zcr_mean` ← mean(zcr)

8.3: `zcr_var` ← variance(zcr)
8.4: append `zcr_mean`, `zcr_var` to `f`
9: Extract tonnetz
9.1: `tonnetz` ← tonnetz(y)
9.2: `tonnetz_mean` ← mean(tonnetz)
9.3: `tonnetz_var` ← variance(tonnetz)
9.4: append `tonnetz_mean`, `tonnetz_var` to `f`
10: Extract Mel-spectrogram
10.1: `S` ← mel_spectrogram(y)
10.2: `mel` ← amplitude_to_dB(S)
10.3: `mel_mean` ← mean(mel)
10.4: `mel_var` ← variance(mel)
10.5: append `mel_mean`, `mel_var` to `f`
11: Extract MFCC
11.1: `mfcc` ← MFCC(y)
11.2: For each coefficient `c` in `mfcc`:
a. `mfcc_mean[c]` ← mean(c)
b. `mfcc_var[c]` ← variance(c)
c. append `mfcc_mean[c]`, `mfcc_var[c]` to `f`
12: Return array `f`

2.3. Emotion Annotator Criteria

This study followed several criteria to find suitable annotators. The annotator needed to be an Indonesian citizen to avoid geographic perception bias. Although any gender could participate, gender bias was disregarded due to time constraints. The minimum age requirement was set at 18 to ensure emotional judgment stability. Additionally, annotators with no hearing problems were preferred to ensure accurate emotion annotation. While annotators could appreciate various music genres, a preference for Javanese music was ideal. Finally, having music experiences, such as being a music director, sound editor, choir member, or instrumentalist, helped avoid misinterpretation of emotions.

Although based on the Self-Assessment Manikin (SAM) model, the questionnaire used a 9-point Likert scale for valence and arousal dimensions. The dominance dimension was excluded, following common MER practices that prioritize the valence-arousal plane for affective mapping.

2.4. Data Collection

The dataset being collected was downloaded based on a similar song title on YouTube using the Spotify package. Due to copyright considerations, the raw audio files cannot be distributed. Using Google Forms, as shown in **Figure 6**, the emotion annotator, based on the Arousal-Valence model based on the SAM model, with nine scale points (Cheetham, Wu, Pauli, & Jancke, 2015), manually annotated the music before the audio was merged with the dataset. There are 100 songs collected from various genres of Javanese music and then annotated by annotators. The form was divided into 4 section groups during annotation, each with 25 songs. The annotator has to complete all the questionnaires given after listening to the song, based on the list of songs given before.

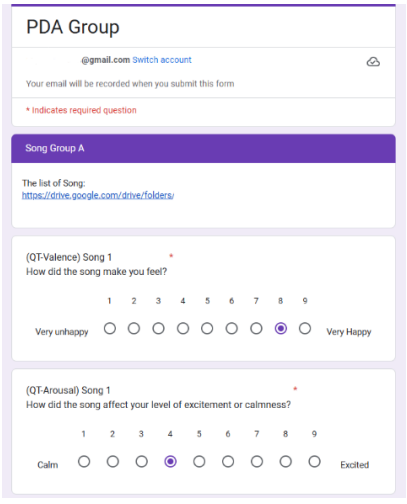

Figure 6 Google Form for Annotation Process

Since the scale followed the SAM model, normalizing it to 0 to 1 was needed to make plotting emotion easier in the Thayer model (Thayer, 1990) and Russell (1980).

2.5. Machine Learning Classifier

Three different machine learning classifiers with separate configurations are used in this work. Support Vector Machine (SVM) is the first choice. A widely used machine learning classifier, Support Vector Machine (SVM), helps solve pattern identification problems (Burges, 1998). Sometimes known as determining the "hyperplane," the optimization constraint was developed utilizing a set of mathematical equations to address the problem. SVM allows one to use kernels in concert with the SVM technique (Salcedo-Sanz et al., 2014). Calculating the inner product of the new vectors helps one translate the data into a familiar environment. Linear, radial basis functions (RBF), and polynomial equations rank the kernels most commonly used.

In machine learning, 1D-CNN is the second classifier used in the field. Deep learning classifiers are a subset of Convolutional Neural Networks (CNN), 1D-CNN The multi-layered artificial neural network (ANN) is referred to as "deep learning" (Dutta et al., 2024). A fully connected layer, a non-linearity layer, a pooling layer, and a convolutional layer are just a few of the numerous kinds of layers a convolutional neural network (CNN) might comprise. CNN, often known as Convolutional Neural Network, has become somewhat well-known for its extraordinary capacity to solve machine learning problems, especially related to image processing (Matsumoto et al., 1990). Although LSTM architectures are effective for sequential modeling, 1D-CNN was chosen for its ability to efficiently capture local temporal dependencies in audio data while maintaining lower computational complexity. Prior studies have demonstrated the effectiveness of 1D-CNN in music emotion recognition with limited datasets (Allamy & Koerich, 2021).

XGBoost, often known as Extreme Gradient Boosting, is the third classifier applied in this work. XGBoost is a potent and extensively used machine learning tool well-known for its outstanding accuracy and efficiency (Chen et al., 2015). It is a member of the family of gradient-boosting algorithms, which repeatedly mix many weak learners, usually decision trees, to produce a robust predictive model. XGBoost shines at managing challenging data, using regularizing methods to stop overfitting and maximizing computing performance. These features make it a good fit for emotion identification activities, as complex patterns in audio data call for thorough modeling and effective processing.

The selection of three classifiers aimed to compare performance across distinct model families: kernel-based (SVM), deep learning (1D-CNN), and ensemble (XGBoost). XGBoost itself represents an ensemble approach that combines multiple weak learners. More complex hybrid ensembles were excluded to avoid overfitting, given the limited dataset size.

3. RESULT AND DISCUSSION

This chapter shows the experiment's results and discusses them based on the research explained in the previous chapter. It will explain the results of the emotion annotation process and the classification of emotion recognition and discuss the performance of the model classifier.

3.1. Emotion Annotation Process

All annotations were conducted manually by five qualified annotators who met the inclusion criteria described in Section 2.3. Each annotator independently rated songs on the valence-arousal scale, based on the Thayer model, ensuring culturally accurate representation of emotions.

At the beginning of the annotation process, a Google form with a questionnaire used to identify arousal and valence for each Javanese song, which consisted of two questions per song, was made. The form was created using the SAM model. Emotion annotation involves five people, who are composed of 2 members of the Sampoerna Choir Club, one member of the Sampoerna Summer Club, one church volunteer singer, and one sound man. The total number of annotated songs is 100, and they come from several genres in Javanese music, such as Dangdut, Koplo, and Campur Sari. Five people annotated each song, and the total annotated data will be 500 using the Google form provided.

To make tracking the progress of the annotation music more manageable, the annotation was divided into four groups, each consisting of 25 songs. After all the songs have been annotated, the mean of the A-V score is needed to find the average value of each question. Since the scale value was 1 to 9, the A-V mean score needed to be converted to the scale 0 to 1 to match the plotting of emotion that was used in the Thayer model (Thayer, 1990), (Russell, 1980) by adjusting to the scale 0 to 1 so emotion annotation can be manually annotated. In other word, each song was rated on a 9-point scale for both Arousal and Valence following the SAM model. Scores were normalized to a 0–1 range, and the mean across five annotators was used to obtain the final A–V score per song.

Annotator ratings were analyzed for consistency using variance and interquartile range (IQR), which indicated minimal dispersion. Therefore, mean aggregation was used to preserve subtle differences in perceived emotion. However, median aggregation will be explored in future work for robustness.

3.2. Emotion Recognition Process

Emotion recognition will make use of the dataset following the annotations. Using Librosa to extract the data from the music, 55 features, including chroma, rms, mel, and the MFCC feature (Gupta et al., 2013), were obtained. Because it is a type of file identified by Librosa and is lightweight compared to the MP3 format, the audio file utilized was in the WAV format. Later, the feature will recognize four emotions based on the Thayer model (Thayer, 1990), Russell (1980), and the emotion label given for each audio file.

The classifier that was used for emotion recognition was a 1D-CNN with hyperparameter tuning, with four layers of CNN, consisting of 2 Conv1D and maxpooling layers and two fully connected layers, with an adjusted regularizer and dropout to tackle the overfitting model problem. After the emotion recognition process has been completed, the performance evaluation for each classifier is made to determine which classifier performs better in classifying emotion recognition in Javanese music.

3.3. Classifying Emotion with 3-Model Classifier

With the labelled dataset, classifications can be processed. At first, after the variable is called from the saved file, the variable is scaled as the range of the variables varies distinctly. The features need to be scaled into similar ranges to improve the learning process. For this experiment, a Min-Max scaler (Priyambudi & Nugroho, 2024) and a one-hot encoder (Rodríguez et al., 2018) were used. During the training process, the model result always suffers from overfitting, regardless of adjustments, with an accuracy of 0.6446%, a validation accuracy of 0.433%, and a parameter and number of input features. Train validation loss seems better as the values always decrease, indicating the model

learned something from the dataset, and the errors made by the model decreased.

Initially, the model was built with three activated layers: two dense and one dropout layer. Later, it was found that using three filters in the model would increase learning accuracy but would not align with validation accuracy, causing overfitting. Thus, dropping the third filter seems to help avoid overfitting, even though the cost was low in terms of accuracy and validation accuracy.

Based on the result, it is increasing the result by focusing on increasing validation accuracy using the hyperparameter needed. The result of hyperparameter using GridSearch with objective 'val_accuracy' does not give the best result, although the best val-accuracy obtained was 0.7392% or +0.0925% increase compared to the first model. However, the result is overfitted and thus not suitable for the research.

Then, this study proposed splitting the dataset with train-test ratios of 60:20:20, 70:15:15, and 80:10:10 for train, validation, and test sets, respectively. This research also conducted an experiment using the SVM model and XGBoost in comparison with deep learning at 1D-CNN.

Figures 7, 8, and **9** show train validation on accuracy and loss on those three ratio models with the same classifier, 1D-CNN. At the ratio of 60:20:20, indications of overflight show higher differences in accuracy and validation accuracy, as shown in **Figure 7**. To prevent this overfitting indication, the initial idea was to reduce the number of dropouts so that it would only use one dropout. However, it impacts other classes badly, as shown in **Figure 10**.

The overfitting is shown clearly, as the difference is almost 0.3 points, with the train validation loss not decreasing. In the end, the decision was taken not to change the dropout. The difference in accuracy in the classification report was used as a metric to compare and understand the model's performance, as shown in **Figure 9**.

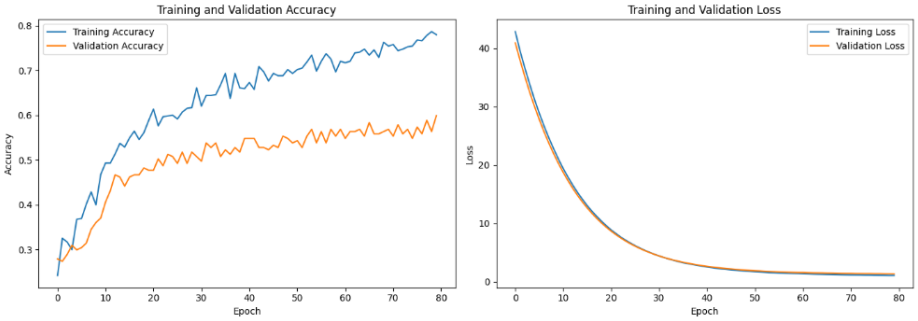

Figure 7 Train validation accuracy and loss of 1D-CNN of 60:20:20

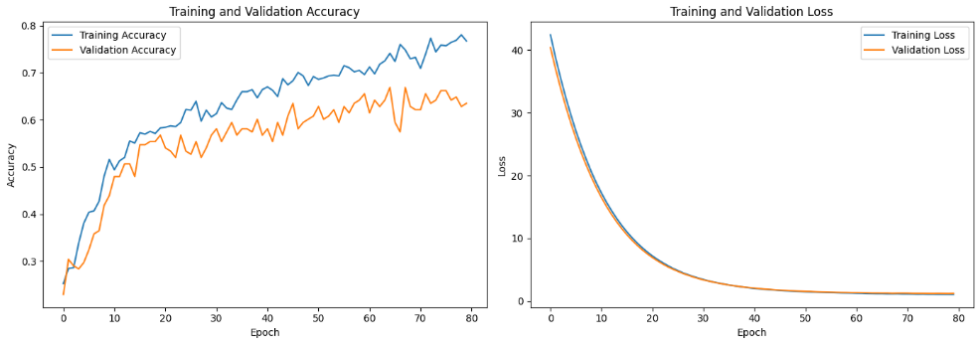

Figure 8 Train validation accuracy and loss of 1D-CNN of 70:15:15

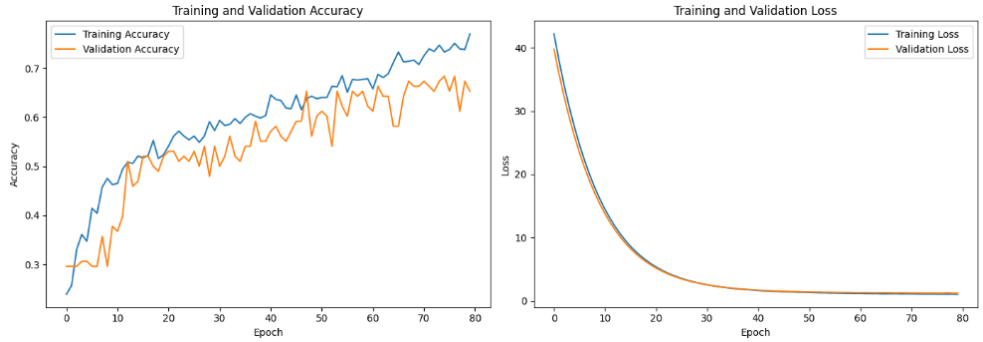

Figure 9 Train validation accuracy and loss of 1D-CNN of 80:10:10

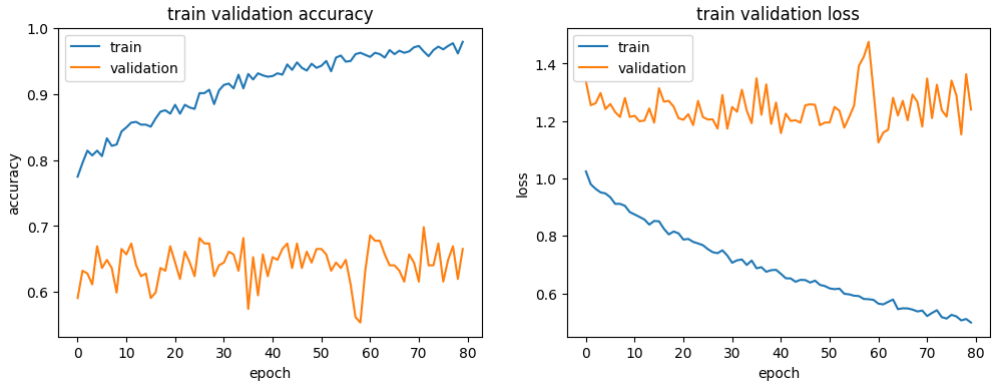

Figure 10 Effect of reducing dropout on train validation accuracy and loss of 1D-CNN of 70:15:15

Table 1 Classification report of Javanese song with ratio 60:20:20 using 1D-CNN

Classifier: 1D-CNN				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.71	0.66	0.68	44
Happy	0.8	0.77	0.79	48
Relaxed	0.68	0.54	0.6	59
Sad	0.54	0.74	0.62	46
Averages				
Accuracy			0.67	197
macro avg	0.68	0.68	0.67	197
weighted avg	0.68	0.67	0.67	197

Table 2 Classification report of Javanese song with ratio 60:20:20 using SVM

Classifier: SVM				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.82	0.7	0.76	44
Happy	0.73	0.9	0.8	48
Relaxed	0.71	0.76	0.74	59
Sad	0.84	0.67	0.75	46
Averages				
Accuracy			0.76	197
macro avg	0.77	0.76	0.76	197
weighted avg	0.77	0.76	0.76	197

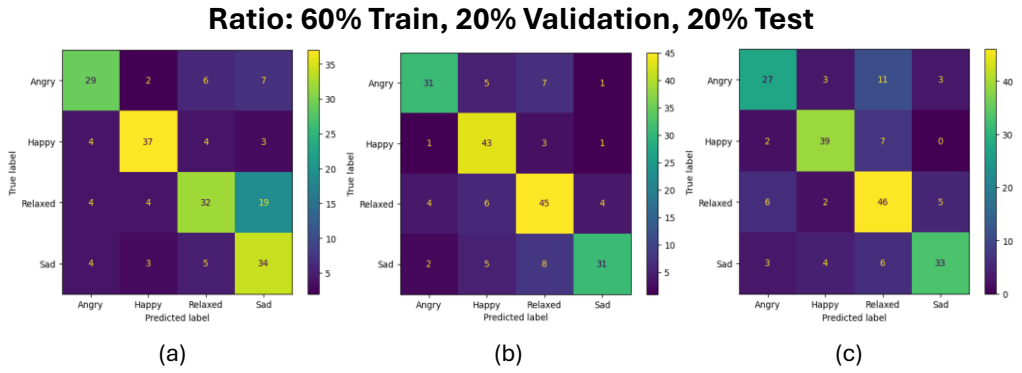

Figure 11 Confusion Matrix of Javanese Song with ratio 60:20:20 using classifier: (a) 1D CNN, (b) SVM, (c) XGBoost

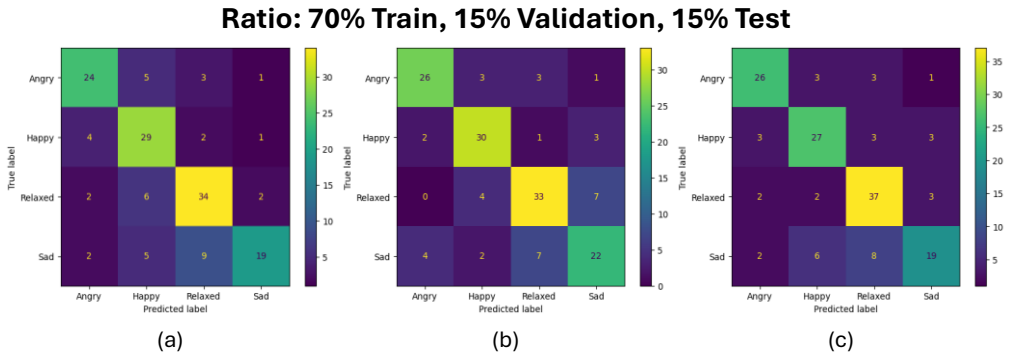

Figure 12. Confusion Matrix of Javanese Song with ratio 70:15:15 using classifier: (a) 1D CNN, (b) SVM, (c) XGBoost

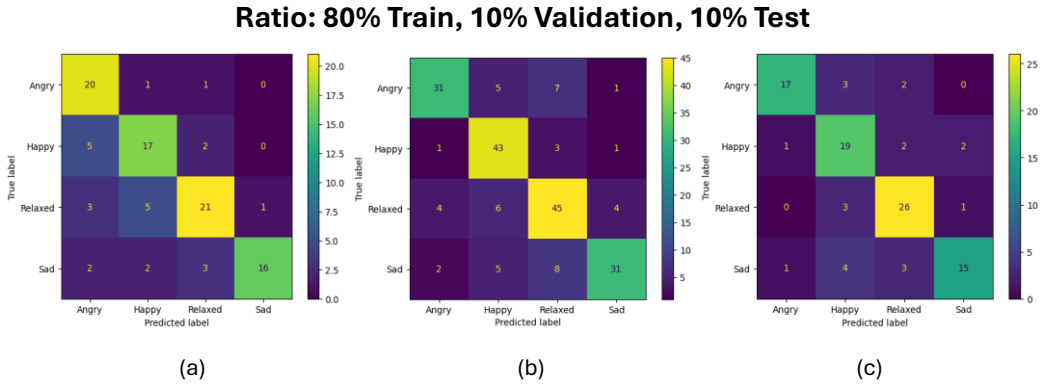

Figure 13 Confusion Matrix of Javanese Song with ratio 80:10:10 using classifier: (a) 1D CNN, (b) SVM, (c) XGBoost

Table 3 Classification report of Javanese song with ratio 60:20:20 using XGBoost

Classifier: XGBoost				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.71	0.61	0.66	44
Happy	0.81	0.81	0.81	48
Relaxed	0.66	0.78	0.71	59
Sad	0.8	0.72	0.76	46
Averages				
Accuracy			0.74	197
macro avg	0.75	0.73	0.74	197
weighted avg	0.74	0.74	0.74	197

Table 4 Classification report of Javanese song with ratio 70:15:15 using 1D-CNN

Classifier: 1D-CNN				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.75	0.73	0.74	33
Happy	0.64	0.81	0.72	36
Relaxed	0.71	0.77	0.74	44
Sad	0.83	0.54	0.66	35
Averages				
Accuracy			0.72	148
macro avg	0.73	0.71	0.71	148
weighted avg	0.73	0.72	0.71	148

Table 5 Classification report of Javanese song with ratio 70:15:15 using SVM

Classifier: SVM				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.81	0.79	0.8	33
Happy	0.77	0.83	0.8	36
Relaxed	0.75	0.75	0.75	44
Sad	0.67	0.63	0.65	35
Averages				
Accuracy			0.75	148
macro avg	0.75	0.75	0.75	148
weighted avg	0.75	0.75	0.75	148

Table 6 Classification report of Javanese song with ratio 70:15:15 using XGBoost

Classifier: XGBoost				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.79	0.79	0.79	33
Happy	0.71	0.75	0.73	36
Relaxed	0.73	0.84	0.78	44
Sad	0.73	0.54	0.62	35
Averages				
Accuracy			0.74	148
macro avg	0.74	0.73	0.73	148
weighted avg	0.74	0.74	0.73	148

Table 7 Classification report of Javanese song with ratio 80:10:10 using 1D-CNN

Classifier: 1D-CNN				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.67	0.91	0.77	22
Happy	0.68	0.71	0.69	24
Relaxed	0.78	0.7	0.74	30
Sad	0.94	0.7	0.8	23
Averages				
Accuracy			0.75	99
macro avg	0.77	0.75	0.75	99
weighted avg	0.77	0.75	0.75	99

Table 8 Classification report of Javanese song with ratio 80:10:10 using SVM

Classifier: SVM				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.82	0.82	0.82	22
Happy	0.71	0.83	0.77	24
Relaxed	0.77	0.67	0.71	30
Sad	0.7	0.7	0.7	23
Averages				
Accuracy			0.75	99
macro avg	0.75	0.75	0.75	99
weighted avg	0.75	0.75	0.75	99

Table 9 Classification report of Javanese song with ratio 80:10:10 using XGBoost

Classifier: XGBoost				
Emotion	Precision	Recall	F1-Score	Support
Angry	0.89	0.77	0.83	22
Happy	0.66	0.79	0.72	24
Relaxed	0.79	0.87	0.83	30
Sad	0.83	0.65	0.73	23
Averages				
Accuracy			0.78	99
macro avg	0.79	0.77	0.78	99
weighted avg	0.79	0.78	0.78	99

Figure 11, Table 1, 2 and **3** illustrate the performance of the three classifiers, 1D-CNN, SVM, and XGBoost, on the Javanese song dataset with a data split of 60% training, 20% validation, and 20% test. The confusion matrices in **Figure 11** indicate that the SVM classifier (b) achieved the most consistent correct classification across all classes, reflected by fewer misclassifications compared to the 1D-CNN (a) and XGBoost (c). This result is confirmed by the classification report in **Table 1, 2** and **3,** where SVM reached a macro-average F1-score of 0.76, slightly higher than both 1D-CNN (0.67) and XGBoost (0.74). This result presents that the SVM's balanced performance across the four emotion categories.

Figure 12, Table 4, 5, and **6** present results under a 70% training, 15% validation, and 15% testing split. Confusion matrices in **Figure 12** suggest that all classifiers improved over the previous ratio, with clearer diagonals indicating better true positive rates. The SVM classifier again demonstrated robust classification, particularly for the “Relaxed” and “Happy” classes, as seen in its confusion matrix (b). Supporting this, the classification report in **Table 4, 5,** and **6** shows SVM achieving a macro-average F1-score of 0.75, which is marginally better than the 1D-CNN (0.71) and comparable to XGBoost (0.73). This reflects that an increased training set improved SVM stability.

In **Figures 13, Table 7, 8,** and **9** which use an 80% training, 10% validation, and 10% test split, performance patterns shift. The confusion matrices in **Figure 13** indicate that 1D-CNN (a) and XGBoost (c) show slightly more confusion among classes, while SVM (b) maintains strong classification results, with most true labels correctly predicted. The classification reports in **Tables 7, 8,** and **9** support this finding. It shows consistent macro-average F1-scores of 0.75 for both 1D-CNN and SVM, and 0.78 for XGBoost. This result shows that XGBoost benefited from the larger training set. SVM still worked well with a smaller test set that showed stable results.

3.4. Performance Analysis

The performance of three machine learning models, 1D-CNN, SVM, and XGBoost, is investigated here for emotion identification in Javanese music. Based on the Thayer model (Thayer, 1990) and Russell's (1980) model, the models were trained and assessed using a human-annotated dataset of one hundred Javanese

songs, split into four moods (angry, sad, happy, and relaxed). Each model's performance was evaluated using data split ratios (60:20:20, 70:15:15, and 80:10:10), reflecting the data distribution for training, validation, and testing.

Table 10 Classifier Performance on several split ratios

Split Ratio	Classification Method	Training		Testing	
		Accuracy	Precision	Accuracy	Precision
60:20:20	1D-CNN	0.59	0.60	0.67	0.68
	SVM	0.57	0.60	0.76	0.77
	XGBoost	0.70	0.72	0.75	0.74
70:15:15	1D-CNN	0.63	0.65	0.73	0.72
	SVM	0.62	0.63	0.75	0.75
	XGBoost	0.75	0.74	0.74	0.74
80:10:10	1D-CNN	0.65	0.67	0.77	0.75
	SVM	0.61	0.64	0.75	0.75
	XGBoost	0.76	0.76	0.78	0.79

Table 10 presents a comparative analysis of classifier performance across three different data split ratios for training and testing. The results show that XGBoost consistently achieved the highest training and testing accuracy among the evaluated models, reaching up to 0.76 accuracy in training and 0.78 accuracy in testing under the 80:10:10 split. In contrast, 1D-CNN demonstrated moderate performance that improved from 0.59 training accuracy with the 60:20:20 split to 0.65 with the 80:10:10 split, while its testing accuracy also increased from 0.67 to 0.77. SVM displayed the most stable testing accuracy across all splits. Despite its lower training accuracy, it maintains around 0.75 to 0.76. Furthermore, precision scores were relatively consistent for each classifier, with XGBoost achieving the highest testing precision of 0.79 in the 80:10:10 configuration. These findings indicate that XGBoost benefits most from a larger training set, while SVM remains consistent regardless of data split, and 1D-CNN improves gradually as more data is available for training.

4. CONCLUSION

This study explored the application of machine learning techniques to recognize emotions in Javanese music, an area that remains underrepresented in existing Music Emotion Recognition (MER) research. By employing a manually annotated dataset of 100 Javanese songs and extracting key audio features using MFCC and related descriptors, the study assessed the performance of three classification models: 1D Convolutional Neural Networks (1D-CNN), Support Vector Machines (SVM), and XGBoost.

The results demonstrated that all three models could achieve classification accuracy above 70%, with SVM and XGBoost consistently outperforming 1D-CNN in terms of stability and precision. In particular, SVM achieves the most stable performance across various data split ratios. In contrast, XGBoost achieves the highest accuracy and precision at an 80:10:10 split. This means that access to more data can further improve model performance.

The study demonstrates that successful MER in non-Western music requires culturally appropriate datasets and well-tuned machine learning models. A successful machine learning application in this context advances the field of computational ethnomusicology and opens new possibilities for emotion-aware music applications tailored to diverse cultural expressions. Future work could expand the dataset, explore additional deep learning architectures, or incorporate multimodal data to enhance emotion recognition in traditional music

REFERENCES

Allamy, S., & Koerich, A. L. (2021, December). 1D CNN architectures for music genre classification. In 2021 IEEE symposium series on computational intelligence (SSCI) (pp. 01-07). IEEE.

Ansari, M. R., Tumpa, S. A., Raya, J. A. F., & Murshed, M. N. (2021, September). Comparison between support vector machine and random forest for audio classification. In 2021 International Conference on Electronics, Communications and Information Technology (ICECIT) (pp. 1-4). IEEE.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

Cheetham, M., Wu, L., Pauli, P., & Jancke, L. (2015). Arousal, valence, and the uncanny valley: Psychophysiological and self-report findings. Frontiers in psychology, 6, 981.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., ... & Zhou, T. (2015). Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4), 1-4.

Dutta, D., Porel, S., Tah, D., & Dutta, P. (2024). One-Dimensional Convolutional Neural Network for Data Classification. In Advanced Technologies for Realizing Sustainable Development Goals: 5G, AI, Big Data, Blockchain, and Industry 4.0 Application (pp. 37-62). Bentham Science Publishers.

Gupta, S., Jaafar, J., Ahmad, W. W., & Bansal, A. (2013). Feature extraction using MFCC. Signal & Image Processing: An International Journal, 4(4), 101-108.

Kaur, J., & Kumar, A. (2021). Speech emotion recognition using CNN, k-NN, MLP and random forest. In Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020 (pp. 499-509). Springer Singapore.

Matsumoto, T., Yokohama, T., Suzuki, H., Furukawa, R., Oshimoto, A., Shimmi, T., ... & Chua, L. O. (1990, December). Several image processing examples by CNN. In IEEE International Workshop on Cellular Neural Networks and their Applications (pp. 100-111). IEEE.

Naveenkumar, M., & Kaliappan, V. K. (2019, November). Audio based emotion detection and recognizing tool using mel frequency based cepstral coefficient. In Journal of Physics: Conference Series (Vol. 1362, No. 1, p. 012063). IOP Publishing.

Nissar, I., Rizvi, D. R., Masood, S., & Mir, A. N. (2019). Voice-Based Detection of Parkinson's Disease through Ensemble Machine Learning Approach: A Performance Study. EAI Endorsed Trans. Pervasive Health Technol., 5(19), e2.

Passricha, V., & Aggarwal, R. K. (2019). A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition. Journal of Intelligent Systems, 29(1), 1261-1274.

Priyambudi, Z. S., & Nugroho, Y. S. (2024, January). Which algorithm is better? An implementation of normalization to predict student performance. In AIP Conference Proceedings (Vol. 2926, No. 1, p. 020110). AIP Publishing LLC.

Rezaul, K. M., Jewel, M., Islam, M. S., Siddiquee, K. N. E. A., Barua, N., Rahman, M. A., ... & Asha, U. F. T. (2024). Enhancing Audio Classification Through MFCC Feature Extraction

and Data Augmentation with CNN and RNN Models. International Journal of Advanced Computer Science and Applications, 15(7), 37-53.

Rodríguez, P., Bautista, M. A., Gonzalez, J., & Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. Image and Vision Computing, 75, 21-31.

Russell, J. A. (1980). A circumplex model of affect. Journal of personality and social psychology, 39(6), 1161.

Salcedo-Sanz, S., Rojo-Álvarez, J. L., Martínez-Ramón, M., & Camps-Valls, G. (2014). Support vector machines in engineering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(3), 234-267.

Scherer, K. R., Clark-Polner, E., & Mortillaro, M. (2011). In the eye of the beholder? Universality and cultural specificity in the expression and perception of emotion. International Journal of Psychology, 46(6), 401-435.

Sloboda, J. A., & Juslin, P. N. (2001). Psychological perspectives on music and emotion. Music and emotion: Theory and research, 71-104.

Thayer, R. E. (1990). The biopsychology of mood and arousal. Oxford University Press.

Yang, Y.-H., & Chen, H. H. (2011). Music emotion recognition (1st ed.). CRC Press. https://doi.org/10.1201/b10731